Frontier overview

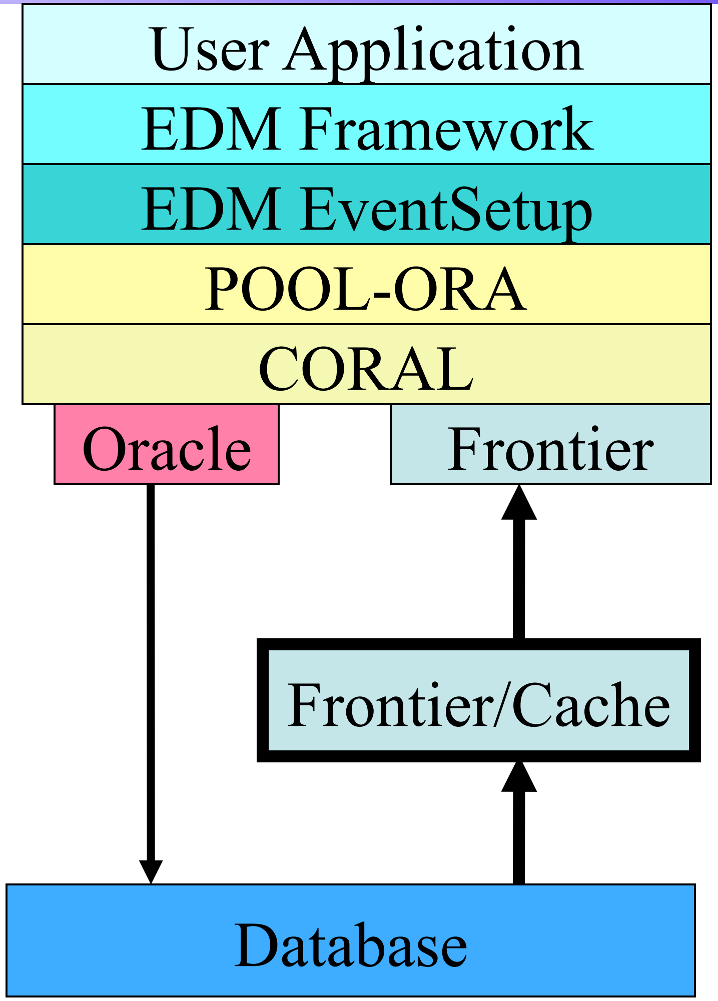
Grid Facilities Department meeting
Dave Dykstra

Motivation

- CMS ***conditions data*** includes calibration, alignment, and configuration information used for offline detector ***event data*** processing.
- Conditions data is keyed by time (run number) and defined to be immutable, i.e. new entries require new tags (versions).
- Caching such info close to the processing activity provides significant performance gains.
- Readily deployable, highly reliable and easily maintainable web proxy/caching servers are a logical solution.

CMS Software Stack

- POOL-ORA (Object Relational Access) is used to map C++ objects to Relational schema.
- A CORAL-Frontier plugin provides read-only access to the POOL DB objects via Frontier.



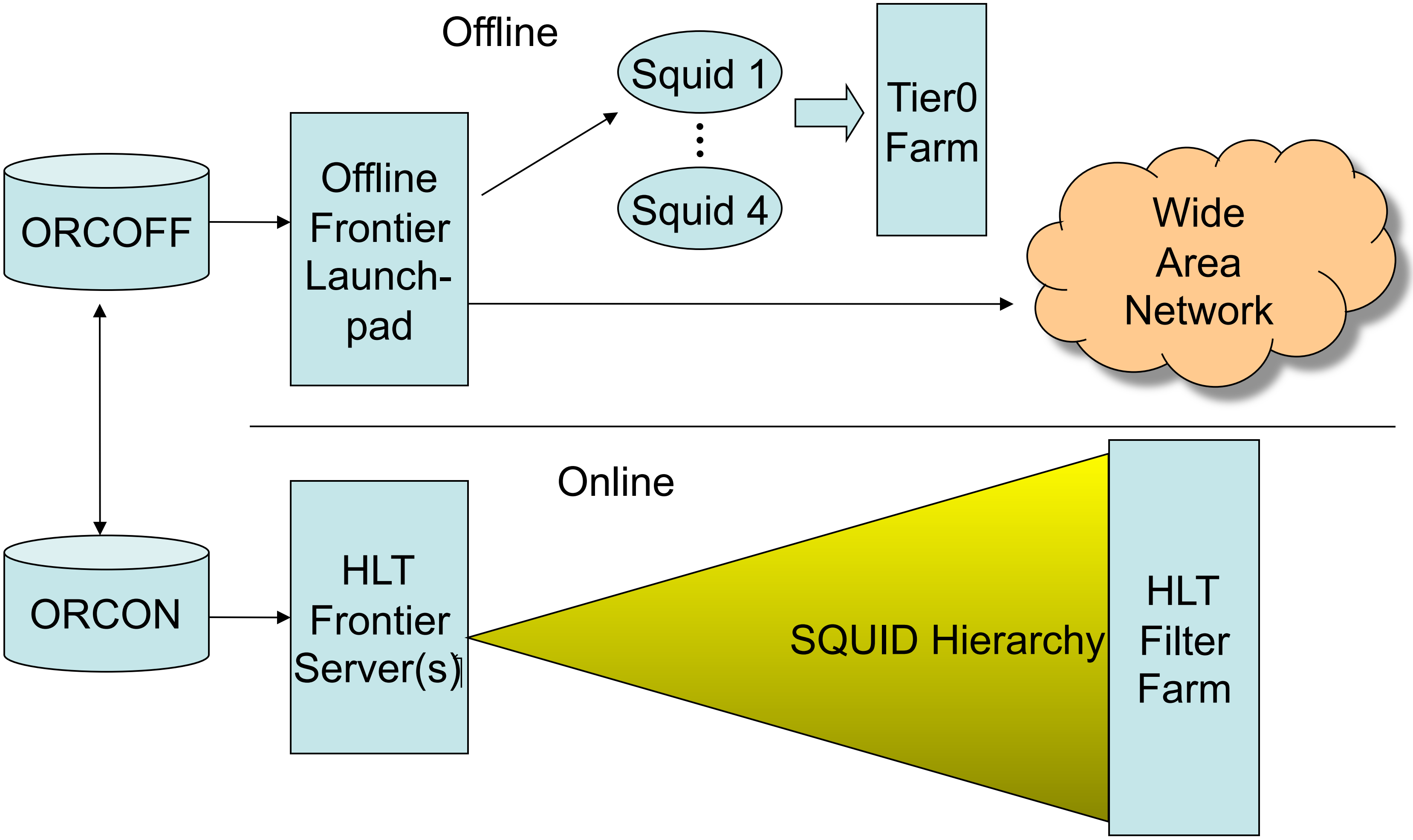
Implementation

- POOL and CORAL generate SQL queries from the CMSSW C++ objects.
- The Frontier client converts the SQL into an HTTP GET and sends it over the network to the Frontier server.
- The Frontier server, a servlet under Tomcat, unpacks the SQL request, sends it to the DB server, and retrieves the needed data.
- The data is optionally compressed, and then packed into an HTTP formatted stream sent back to the client.
- Squid proxy/caching server(s) between the Frontier server and client caches requested objects, significantly improving performance and reducing load on the DB.

Advantages

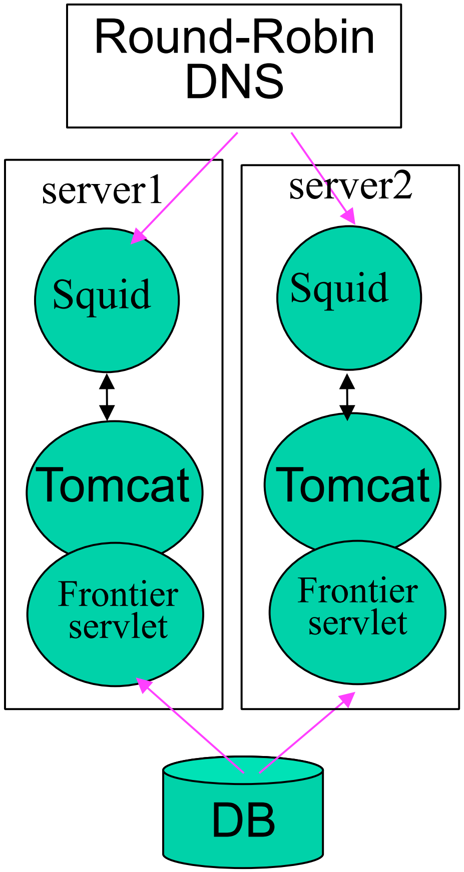
- The system uses standard tools
  - – Highly reliable – Tomcat, Squids well proven
  - – Easy installation – Tar ball and script
  - – Highly configurable – Customize to site environment, security, etc.
  - – Well documented – Books and web sites
  - – Readily monitored - SNMP + MRTG
- Easy administration at Tier-N centers
  - – No DBA's needed beyond central DB @CERN
  - – Caches are loaded on demand and self managing

Overview


Offline
Squid 1
Squid 4
Tier0 Farm
ORCOFF
Offline Frontier Launch-pad
Wide Area Network
Online
ORCON
HLT Frontier Server(s)
SQUID Hierarchy
HLT Filter Farm

Frontier “Launchpad”

- Squid caching proxy
  - Load shared with Round-Robin DNS
  - Configured in “accelerator mode”
  - “Wide open frontier”*
  - “Collapsed forwarding” serializes simultaneous identical requests
- Tomcat - standard
- Frontier servlet
  - Distributed as “war” file
    - Unpack in Tomcat webapps dir
    - Change 2 files if name is different
  - One xml file describes DB connection



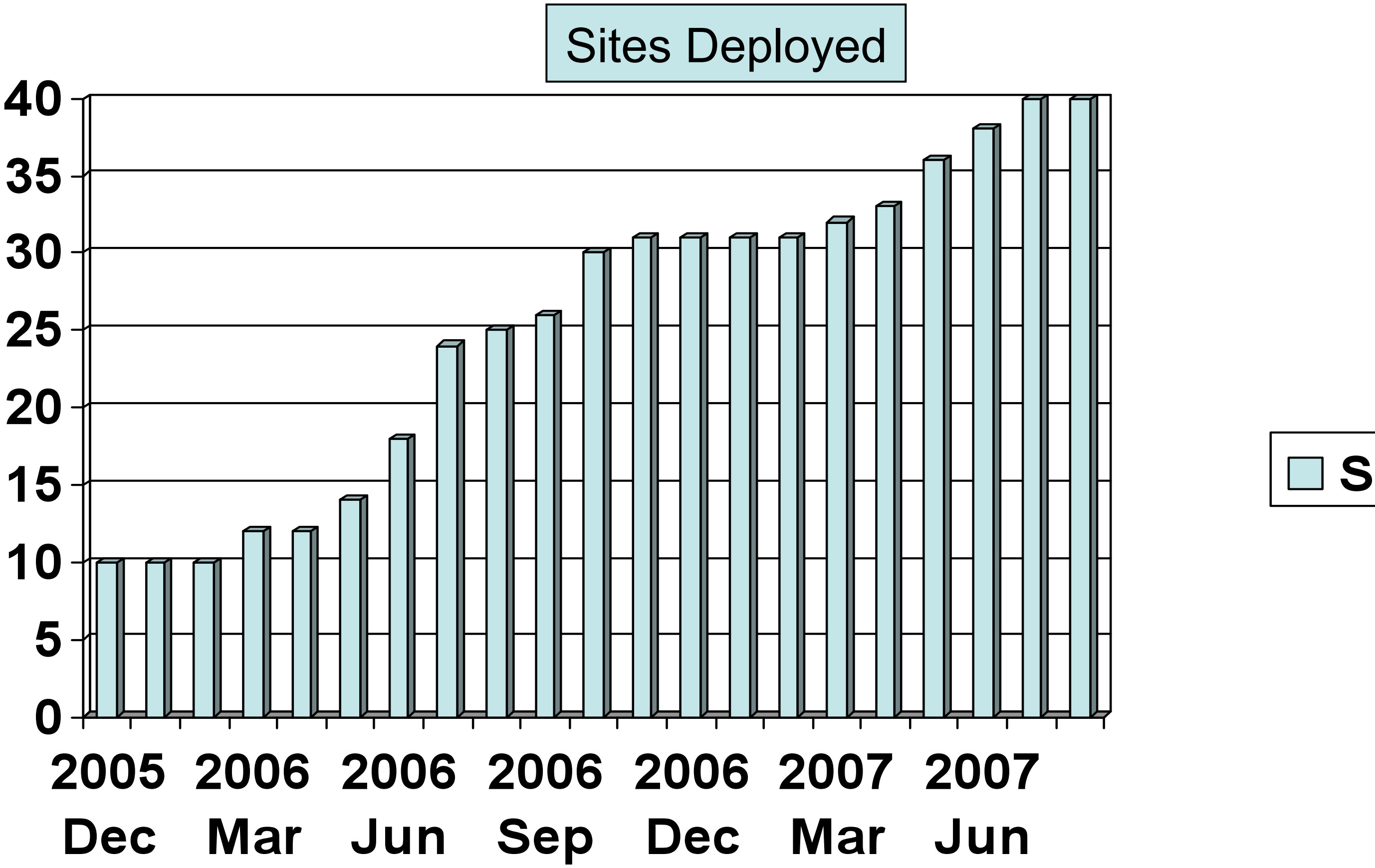
•The squids in the launchpad ONLY talk to the Frontier Tomcat servers. No registration or ACL’s required.

Cache Coherency

- After considering many ideas, adopted following policy:
  - All cached objects have expiration times
  - Shortlived: Metadata objects, including the pointers to payload objects, expire on a short time period
  - Longlived: Payload objects have a long expiration time.
- The value of the short and long expirations is controlled at the Frontier servlet, so can easily be tuned as needed.
- The values of the short and long times varies depending on how the data is being used:
  - Online: the calibrations change quickly as new data is added for upcoming runs so short expiration time is about 10 minutes
  - Tier 0: calibrations change on the order of a few hours as new runs appear for reconstruction so again short time is brief
  - Tier 1 +: Conditions very stable, short time is nightly
  - Development: Data itself may change, short & long both nightly

Squid Deployment Status

- Late 2005, 10 centers used for testing
- Additional installation May through Oct. 2006 used for CSA06
- Additional 20-30 sites for CSA07 possible
- Very few problems with the installation procedures CMS provides.



Frontier for HLT & Tier0

- HLT
  - Startup time for Cal/Ali < 10 seconds.
  - Simultaneous
  - Uses hierarchy of squid caches
- Tier0
  - Startup time for Cal/Ali < 1% of total job time.
  - Usually staggered
  - DNS Round Robin should scale to 8 squids

Parameter	**HLT**	**Tier0**
# Nodes	1000	1000
# Processes	~8k	~3k
Startup	<10 sec all clients	<100 sec per client
Client Access	Simultaneous	Staggered
Cache Load	< 1 Min.	< 1 Min
Tot Obj Size	150 MB*	150 MB*
New Objects	100% / run*	100% / run*
# Squids	1 per node	Scalable (2-8)

* Worst case scenario

Starting many jobs at once - problem

- CMS HLT application has very tight requirements:
  - All nodes start same application at the same time
  - Pre-loading data must be < 1 minute
  - Loading data to jobs must be < 10 seconds
  - Estimating 100MB of data, 2000 nodes, 8 jobs/node
    - 100 * 2000 * 8 = 1.6TB
  - Asymmetrical network
    - Nodes organized in 50 racks of 40 nodes each
    - non-blocking gigabit intra-rack, gigabit inter-rack

Starting many jobs at once - solution

- Solution for CMS HLT: squid on every node
  - Configured to pre-load simultaneously in tiers
  - Each squid feeding 4 means 6 tiers for 2000 nodes
    - 50 racks reached in 3 tiers, 3 tiers inside each rack
  - Measurements on test cluster indicate requirements can be met
    - bottleneck becomes the conversion from DB to http
    - 10-second loading always reads from pre-filled local squid

Summary

- FroNTier is used by CMS for all access to conditions data.
- The ease of deployment, stability of operation, and high performance make the FroNTier approach well suited to the GRID environment being used for CMS offline, as well as for the online environment used by the CMS High Level Trigger (HLT).
- The use of standard software, such as Squid and various monitoring tools, make the system reliable, highly configurable and easy to maintain.
- We have gained significant operational experience over the last year in CMS, there are currently 40 squid sites being monitored and many more expected.